ELSEVIER

Discrete Applied Mathematics 82 (1998) 79–91

DISCRETE APPLIED MATHEMATICS

Polyhedral analysis and decompositions for capacitated plant location-type problems

Bintong Chen[a,*], Monique Guignard[b,1]

[a] *Department of Management and Systems, College of Business and Economics, Washington State University, Pullman, WA 99164-4736, USA*

[b] *Operations and Information Management Department, The Wharton School, University of Pennsylvania, Philadelphia, PA 19104-6366, USA*

Received 21 August 1996; received in revised form 8 July 1997; accepted 28 July 1997

Abstract

We study the polyhedral structure of two primal relaxations of a class of specially structured mixed integer programming problems. This class includes the generalized capacitated plant location problem and a production scheduling problem as special cases. We show that for this class of problems two polyhedra constructed from the constraint sets in two different primal relaxations are identical. The results have the following surprising implications; with linear or nonlinear objective functions, the bounds from two a priori quite different primal relaxations of the capacitated plant location problem are actually equal. In the linear case, this means that a simple Lagrangean substitution yields exactly the same strong bound as the computationally more expensive Lagrangean decomposition introduced in Guignard and Kim (1987) and studied in Cornuejols et al. (1991). © 1998 Elsevier Science B.V. All rights reserved.

1. Introduction

In this paper, we study a class of specially structured mixed integer programs related to but more general than the generalized capacitated plant location problem (GCPLP). We obtain nice polyhedral properties by exploiting the special structure of the problem. We then apply the result to the capacitated plant location problem (CPLP) and show how it can be used to generate efficiently a strong lower bound .

Lagrangean relaxation (LR, see [5]) and Lagrangean decomposition (LD, see [6,10–14] for a general discussion) have primal equivalents, i.e. primal problems which yield the same bounds. Lagrangean decomposition results from creating copies of some variables in some part of the constraints and dualizing the copy constraints. This

* Corresponding author. E-mail: chenbi@wsu.edu.

[1] Research partially supported by NSF grant DDM-9014901.

0166-218X/98/$19.00 © 1998 Elsevier Science B.V. All rights reserved
PII S0166-218X(97)00096-6

artificially creates a staircase structure in the model prior to dualization and decomposes the model into independent submodels after the dualization. The corresponding Lagrangean relaxations would dualize all but one subset of constraints. Lagrangean substitution (LS, see, for instance, [16, 7] for a general description or [9] for an application) induces decomposition by creating more sophisticated substitutions than just copies. In the nonlinear objective function case, one can similarly define primal relaxations without using Lagrangean multipliers [15, 8].

Many Lagrangean decomposition schemes provide better bounds than the corresponding Lagrangean relaxations, as they retain more information on the original problem. However, they also involve more work; at least one more subproblem must be solved than in the corresponding LR. Lagrangean substitution, in general, yields bounds which are in between the corresponding LR and LD bounds. Since they do not create identical copies, but "rough" copies, prior to dualization, some information is understandably lost, and the LS bounds tend to be weaker than the corresponding LD bounds. This is why the result we obtained is surprising. For CPLP, the primal (or primal-equivalent in the linear case) relaxations corresponding to a simple substitution (defined in [7]) and to a computationally more expensive decomposition introduced in [10] and studied in [3] are equivalent. This provides a computationally feasible alternative for computing the strong bound of [10].

The paper is organized as follows. In Section 2, we introduce the generic problem and show that the GCPLP and a dynamic production scheduling problem are special cases. In Section 3, we show that two polyhedra constructed differently from the constraint set of the problem are actually identical. In Section 4, we apply the result to CPLP and show that the simple Lagrangean substitution and the computationally more expensive Lagrangean decomposition mentioned above actually provide the same Lagrangean dual bounds. In Section 5, we apply the result to CPLP with a nonlinear objective function. We show that two primal relaxation bounds are identical.

We now introduce the notation used in the paper. Let I be an index set with cardinality $|I|$. z_I is a column vector with entries z_i, $i \in I$. When there is no ambiguity, the index set I is often dropped from a vector. In that case, e means a vector of ones of appropriate dimension; 0 may mean a scalar zero or a vector of zeros of appropriate dimension; any other vector z is meant to be a vector of full dimension. Let $(S_1),\dots,(S_n)$ represent sets of equality or inequality constraints. We use the notation $P(S_1 \dots S_n)$ to denote the feasible region defined by the constraints $(S_1),\dots,(S_n)$ and $Co(S_1 \dots S_n)$ to denote the convex hull of the corresponding region.

2. A Class of specially structured integer programs

Let N_j, $j = 1,\dots,m$, be mutually exclusive index sets, each with dimension n_j and

$$\bigcup_{j=1}^{m} N_j = N, \qquad \sum_{j=1}^{m} n_j = n.$$

We study the following mixed integer program, called GP later on:

$$\begin{array}{lll} \text{minimize} & f(x,y,s,t), & \\ \text{subject to} & & \\ & \text{(D)} & (x,y,s) \in S_{x,y,s}, \\ & \text{(C)} & d_{N_j}^T x_{N_j} \leqslant a_j y_j \ \forall j, \\ & \text{(T)} & (y,t) \in S_{y,t}, \\ & \text{(N)} & 0 \leqslant x \leqslant e, \\ & \text{(I)} & y_j = 0,1 \ \forall j. \end{array}$$

In the above formulation, x and y are vectors of dimension n and m, respectively. s and t are vectors of any finite dimension. f is a function of variables x, y, s, and t. $S_{x,y,s}$ and $S_{y,t}$ are arbitrary constraint sets of (x,y,s) and (y,t), respectively. d_{N_j} and a_j are assumed to be nonnegative (known) vectors and scalars, respectively. Notice that constraints (C), (N), (I), and the nonnegativity assumption of d_{N_j} and a_j imply that

$$\text{(B)}\ x_{N_j} \leqslant y_j e \quad \forall j.$$

Using the notation introduced in Section 1, we can write GP as

$$\min\{ f(x,y,s,t) \mid (x,y,s,t) \in P(DCTNIB)\}.$$

Although GP has a very special structure, it contains some well-studied problems as special cases. In what follows, we show that GCPLP and a dynamic production planning model are both special cases of GP. We briefly describe both problems and show that they fit into the structure of GP.

GCPLP has been extensively studied in the literature. It includes both CPLP and the capacitated P-median problem as special cases. The integer programing formulation of GCPLP is as follows:

$$\begin{array}{lll} \text{minimize} & f(x,y) = \sum_i \sum_j c_{ij} x_{ij} + \sum_j f_j y_j & \\ \text{subject to} & & \\ & \text{(D)} & \sum_j x_{ij} = 1 \quad \forall i, \\ & \text{(C)} & \sum_i d_i x_{ij} \leqslant a_j y_j \quad \forall j, \\ & \text{(T)} & \sum_j a_j y_j \geqslant \sum_i d_i, \\ & \text{(N)} & 0 \leqslant x_{ij} \leqslant 1 \quad \forall i,j, \\ & \text{(I)} & y_j = 0,1 \quad \forall j, \\ & \text{(B)} & x_{ij} \leqslant y_j \quad \forall i,j, \\ & \text{(T}') & \sum_j y_j = K, \end{array}$$

where constraints (T) and (B) are derived from other constraints. In the above formulation, $i = 1, \ldots, n$ indexes the clients to be served and, $j = 1, \ldots, m$, the potential sites for the plants; c_{ij} is the transportation cost between client i and site j ; f_j is the setup cost of locating a plant at site j; d_i is the demand of client i, and a_j is the capacity of the plant at site j. K is the number of plants to be opened. GCPLP opens K plants, satisfies all customer demands and minimizes total cost while meeting the capacity requirement of each plant. To see that GCPLP fits in the structure of GP, observe that

$$S_{x,y,s} = \left\{ x \,\middle|\, \sum_j x_{ij} = 1 \;\; \forall i \right\} \quad \text{and} \quad S_{y,t} = \left\{ y \,\middle|\, \sum_j a_j y_j \geqslant \sum_i d_i, \;\; \sum_j y_j = K \right\}.$$

The following dynamic production planning model for a tile manufacturing company was described in [2], which is an extension of a production model studied in [4]. A set of products N need to be scheduled on several non-identical flexible production lines L. For each product $i \in N$, L_i is a set of production lines capable of producing the product. A setup cost q_{ij} is incurred when a production line j starts producing a product i after switching from another product, and the production is limited by capacity p_{ij}. The demand for product i in period s is d_{is}. Unmet demands are back-ordered within the planning horizon. C_{ijts} is the unit production and inventory cost if a product i produced on production line j in period t is used to meet demand in period s. The objective is to minimize the total cost during the planning horizon. We refer to [2] for a more detailed description of the model. After some modifications to the original model, we can formulate the above production model as an integer program

$$\begin{aligned}
&\text{minimize} && \sum_{j \in L} \sum_{i \in N_j} \sum_{t=1}^{T} \left\{ \sum_{s=1}^{T} C_{ijts} x_{ijts} + q_{ij} z_{ijt} \right\} \\
&\text{subject to} && \\
&(1) && \sum_{j \in L_i} \sum_{t \in T} x_{ijts} = 1 \quad \forall i \in N, \; s \in T, \\
&(2) && \sum_{s=1}^{T} d_{is} x_{ijts} = p_{ij} y_{ijt} \quad \forall i \in N, \; j \in L_i, \; t \in T, \\
&(3) && \sum_{i \in N_j} y_{ijt} = 1 \quad \forall j \in L, \; t \in T, \\
&(4) && z_{ijt} \geqslant y_{ijt} - y_{ij,t-1} \quad \forall i \in N, \; j \in L_i, \; t \in T, \\
&(5) && 0 \leqslant x_{ijts} \leqslant 1 \quad \forall i \in N, \; j \in L_i, \; t \in T, \; s \in T, \\
&(6) && z_{ijt} = 0, 1 \quad \forall i \in N, \; j \in L_i, \; t \in T, \\
&(7) && y_{ijt} = 0, 1 \quad \forall i \in N, \; j \in L_i, \; t \in T,
\end{aligned}$$

where x_{ijts} is the number of units of product i produced on line j in period t to meet demand in period s; $y_{ijt} = 1$ if product i is produced on production line j in period t and 0 otherwise; $z_{ijt} = 1$ (0) if line j starts producing product i in period t (if in period t line j continues producing product i or produces some other product).

To see that this production model fits in the structure of GP, we observe the following correspondences: (D) = (1); (B) = $x_{ijts} \leqslant y_{ijt}$ for all $i \in N$, $j \in L_i$, t and s; (C) = (2); (T) = (3), (4), (6); (N) = (5); and (I) = (7).

3. Two identical polyhedra

In this section, we exploit the special structure of GP and present the main result of the paper: two polyhedra, D_1 and D_2, constructed in different ways are actually identical. We first define the two polyhedra using the constraints of GP.

$$D_1 = \left\{ (x,y,s,t) \,\middle|\, \begin{array}{ll} (x,y,s) \in Co(DBNI), & (x',y',t) \in Co(CTNI), \\ (x,y) = (x',y'), & \end{array} \right\}$$
$$= \{(x,y,s,t) \mid (x,y,s) \in Co(DBNI), \quad (x,y,t) \in Co(CTNI)\};$$
$$D_2 = \left\{ (x,y,s,t) \,\middle|\, \begin{array}{ll} (x,y,s) \in Co(DBNI), & (x',y',t) \in Co(CTNI), \\ d_{N_j}^T x_{N_j} = d_{N_j}^T x'_{N_j} \quad \forall j, & y = y' \end{array} \right\}.$$

We need the following lemma to show the equivalence of D_1 and D_2.

Lemma 1.

$$\left.\begin{array}{ll} \text{(H1)} & (x',y',t) \in Co(CTNI) \\ \text{(H2)} & 0 \leqslant x_{N_j} \leqslant y_j e \leqslant e \quad \forall j \\ \text{(H3)} & d_{N_j}^T x_{N_j} = d_{N_j}^T x'_{N_j} \quad \forall j \\ \text{(H4)} & y = y' \end{array}\right\} \Rightarrow (x,y,t) \in Co(CTNI).$$

Proof. By (H1), (x',y') satisfies (C), or $d_{N_j}^T x'_{N_j} \leqslant a_j y'_j$ for all j. Using (H3) and (H4), we have $d_{N_j}^T x_{N_j} \leqslant a_j y_j$ for all j, or (x,y) satisfies (C). Similarly, (H1) and (H4) imply that (y,t) satisfies (T). In addition, (x,y) satisfies (B) and (N) by (H2). Therefore, if y satisfies (I), then $(x,y,t) \in Co(CTNI)$.

Suppose now that $y = y'$ has some fractional coordinates. By (H1), there exist $(x'^k, y'^k, t^k) \in P(CTNI)$ and $\alpha^k \in (0,1)$, $\sum_k \alpha^k = 1$ such that $(x',y',t) = \sum_k \alpha^k (x'^k, y'^k, t^k)$. Set $y^k = y'^k$. Then y^k satisfies (I) and (T) and $y = y' = \sum_k \alpha^k y^k$. In what follows, we construct x^k such that

$$\text{(a)}\ (x^k, y^k, t^k) \in P(CTNI), \qquad \text{(b)}\ x = \sum_k \alpha^k x^k.$$

Define $\beta = \{j\colon\ y_j = 0\}$ and $\bar{\beta} = \{j\colon\ y_j > 0\}$ (Notice that $y_j^k = 0$ for all $j \in \beta$). For all k, let

$$x_{N_j}^k = \begin{cases} 0 & \text{for all } j \in \beta, \\ x_{N_j} \dfrac{y_j^k}{y_j} & \text{for all } j \in \bar{\beta}. \end{cases}$$

To verify (a), we show that (x^k, y^k, t^k) satisfies constraints (C), (T), (N), (I), respectively. (x^k, y^k) satisfies (C):

If $j \in \beta$, then

$$d_{N_j}^T x_{N_j}^k \leqslant d_{N_j}^T 0 = 0 \leqslant a_j y_j^k = a_j(0) = 0.$$

If $j \in \bar{\beta}$, then

$$d_{N_j}^T x_{N_j}^k = d_{N_j}^T x_{N_j} \frac{y_j^k}{y_j} \leqslant a_j y_j \frac{y_j^k}{y_j} = a_j y_j^k.$$

The inequality is true because (x, y) satisfies (C).

(y^k, t^k) satisfies (T):

$(x'^k, y'^k, t^k) \in P(CTNI)$ implies that (y'^k, t^k) satisfies (T). Since $y^k = y'^k$, (y^k, t^k) satisfies (T).

x^k satisfies (N):

If $j \in \beta$, $x_{N_j}^k = 0$. If $j \in \bar{\beta}$, $x_{N_j}^k = (y_j^k / y_j) x_{N_j} = (x_{N_j}/y_j) y_j^k$. Since $0 \leqslant x_{N_j}/y_j \leqslant e$ by $(H2)$ and $y_j^k = y_j'^k = 0, 1$, we have $0 \leqslant x_{N_j}^k \leqslant e$.

y^k satisfies (I):

y^k satisfies (I) since $y^k = y'^k$ and y'^k satisfies (I).

We now verify (b), $x = \sum_k \alpha^k x^k$. If $j \in \beta$,

$$x_{N_j} = 0 = \sum_k \alpha^k x_{N_j}^k = \sum_k \alpha^k (0) = 0$$

since $x_i^k = y_j^k = 0$ for all $i \in N_j$ and $j \in \beta$. If $j \in \bar{\beta}$,

$$\sum_k \alpha^k x_{N_j}^k = \sum_k \alpha^k \frac{y_j^k}{y_j} x_{N_j} = \frac{x_{N_j}}{y_j} \left(\sum_k \alpha^k y_j^k \right) = \frac{x_{N_j}}{y_j} y_j = x_{N_j}.$$

Combining (a) and (b), we have shown that for each (x, y, t) satisfying $(H1–H4)$ there exist $(x^k, y^k, t^k) \in P(CTNI)$ such that (x, y, t) can be expressed as a convex combination of these points. It follows that $(x, y, t) \in Co(CTNI)$. □

We are now ready to show the equivalence of the two polyhedra.

Theorem 1. $D_1 = D_2$.

Proof. It is clear that $D_1 \subseteq D_2$ since any $(x, y, s, t) \in D_1$ implies $(x, y, s, t) \in D_2$. Therefore, it suffices to show that $D_1 \supseteq D_2$. Given an arbitrary vector $(x, y, s, t) \in D_2$, we have $(x, y, s) \in Co(DBNI)$, which implies $0 \leqslant x_{N_j} \leqslant y_j e \leqslant e$ for all j. In addition, there exists a vector (x', y') such that $(x', y', t) \in Co(CTNI)$, $d_{N_j}^T x_{N_j} = d_{N_j}^T x'_{N_j}$ for all j, and $y = y'$. It follows that all the conditions (H1–H4) in Lemma 1 are satisfied and therefore $(x, y, t) \in Co(CTNI)$. Since $(x, y, s) \in Co(DBNI)$, we have $(x, y, s, t) \in D_1$. □

4. An example when $D_1 \neq D_2$

In this section, we show by an example that Theorem 1 may not hold if some of the structures of GP are altered. In particular, we provide an example such that $D_1 \neq D_2$ if x in constraint (N) is restricted to be a zero–one integer variable.

Table 1
Feasible solutions of $P(CTNI)$

x_{11}	x_{21}	x_{12}	x_{22}
0	0	0	0
1	0	0	0
0	1	0	0
0	0	1	0
1	0	1	0
0	1	1	0

Consider a mixed integer program with the following constraints:

(D) $x_{11}+x_{12}=1, \quad x_{21}+x_{22}=1,$

(C) $x_{11}+2x_{21}\leqslant 2y_1, \quad x_{12}+2x_{22}\leqslant y_2,$

(T) $2y_1+y_2\geqslant 3,$

(N) $x_{ij}=0,1 \ \forall i,j=1,2,$

(I) $y_j=0,1 \ \forall j=1,2,$

(B) $x_{ij}\leqslant y_j \ \forall i,j=1,2.$

The above constraints describe an instance of CPLP where plants 1 and 2 have capacities of 2 and 1 units, respectively, and clients 1 and 2 have demands of 1 and 2 units, respectively. However, constraint (N) is altered slightly by requiring x_{ij} to be a zero–one integer variable instead of a continuous variable as in CPLP. We show that $D_1\neq D_2$ for the above problem.

First, notice that $y_1=y_2=1$ in both D_1 and D_2. Therefore, it suffices to compare the projection of D_1 and D_2 in the x-space, denoted by D_{1x}, D_{2x}, respectively. Similarly, let Co_x be the projection of a convex hull in the x-space. One can verify that Table 1 provides a set of all feasible x_{ij}'s to $P(CTNI)$. By definition, the projected convex hull $Co_x(CTNI)$ can be expressed as convex combinations of the above six feasible solutions. i.e., $Co_x(CTNI)$ is a set of x_{ij} that satisfies

$$\begin{aligned} x_{11}&=\alpha^2+\alpha^5,\\ x_{21}&=\alpha^3+\alpha^6,\\ x_{12}&=\alpha^4+\alpha^5+\alpha^6,\\ x_{22}&=0, \end{aligned} \tag{1}$$

where $\alpha^k\geqslant 0$, $k=1,\ldots,6$, and $\sum_{k=1}^{6}\alpha^k=1$. One the other hand, one can verify that $Co_x(DBNI)$ is a set of x_{ij} that satisfies

$$\begin{aligned} &x_{11}+x_{12}=1,\\ &x_{21}+x_{22}=1,\\ &0\leqslant x_{ij}\leqslant 1 \quad \forall i,j=1,2. \end{aligned} \tag{2}$$

As a result, D_{1x} is a set of x_{ij} that satisfy both systems (1) and (2), while D_{2x} is a set of x_{ij} that satisfy system (2) and

$$\begin{aligned} x_{11} + 2x_{21} &= \alpha^2 + \alpha^5 + 2\alpha^3 + 2\alpha^6, \\ x_{12} + 2x_{22} &= \alpha^4 + \alpha^5 + \alpha^6, \end{aligned} \tag{3}$$

where α^k is defined in (1). By setting $\alpha^k = 0$, $k = 1, \dots 5$, and $\alpha^6 = 1$, one can verify that $(x_{11}, x_{21}, x_{12}, x_{22}) = (1, 0.5, 0, 0.5)$ belongs to D_{2x} but not D_{1x}. It follows that $D_{1x} \subset (\neq) D_{2x}$, and therefore, $D_1 \subset (\neq) D_2$.

5. Two identical dual bounds for CPLP

We now apply the result of the previous section to CPLP, a special case of GP. As a result of Theorem 1, we show that two variable substitution approaches generate the same Lagrangean bound. CPLP differs from GCPLP by removing the requirement that exactly K plants should be opened (constraint (T') in GCPLP). Using the notation introduced in Section 2 for GCPLP, we can formulate CPLP as the following mixed integer program:

$$\min\{f(x, y) \mid (x, y) \in P(DCNIBT)\}.$$

In [3], the authors provided a very interesting theoretical and computational comparison of various approaches for solving CPLP. In particular, they compared lower bounds generated by all Lagrangean relaxations (LRs) and Lagrangean decompositions (LDs) obtained by splitting the constraints of CPLP. Notice that they did not consider the possibility of using the same subset of constraints in several Lagrangean subproblems, i.e. they did not discuss the decompositions in which two or more Lagrangean submodels overlap. Among 25 meaningful LDs, two generate lower bounds that are different from those generated by their corresponding LRs and only one of these two may benefit from the computation. (Other LDs are computationally inferior to the corresponding LRs). To obtain the particular LD bound (we will from now on refer to it as “the” LD bound since it is the only one which makes computational sense), we copy all variables and rewrite CPLP as

$$\min\left\{ f(x, y) \,\middle|\, \begin{array}{l} (x, y) \in P(DBNI), \quad (x', y') \in P(CTNI), \\ (x, y) = (x', y') \end{array} \right\}.$$

The LD is obtained by performing Lagrangean relaxation on the copy constraint $(x, y) = (x', y')$. Let $u \in R^{n \times m}$ and $w \in R^m$ be the multipliers associated with equalities $x = x'$ and $y = y'$, respectively. It is not difficult to show that the Lagrangean subproblem can be simplified as

$$\begin{aligned} &\min\{f(x, y) + u^T x + w^T y \mid (x, y) \in P(DBNI)\} \\ &\quad - \max\{(\gamma(u) + w)^T y' \mid y' \in P(TI)\}, \end{aligned}$$

where $\gamma(u) \in R^m$ is a column vector whose entries are defined by

$$\gamma_j(u) = \max\left\{ \sum_i u_{ij} x_{ij} \,\middle|\, \sum_i d_i x_{ij} \leqslant a_j, 0 \leqslant x_{ij} \leqslant 1 \right\}, \quad j = 1, \ldots, m.$$

Let $V(LD)$ be the lower dual bound generated by the above LD. It has been shown [3] that

$$\max\{Z_C^T, Z_D\} \leqslant V(LD) \leqslant Z_C,$$

where Z_B^A denotes the LR bound obtained by dropping constraint A and dualizing constraint B. Clearly, the lower bound $V(LD)$ is quite strong. However, $(n+1)m$ multipliers need to be adjusted at each iteration while calculating $V(LD)$. In other words, the Lagrangean decomposition dual is an optimization problem in $R^{(n+1)m}$.

A different substitution approach, which aggregates the x_{ij} variables, called LDA (for Lagrangean Decomposition/Aggregation), can be used to calculate a lower bound for CPLP ([7]). The LDA starts by rewriting CPLP as

$$\min\left\{ f(x,y) \,\middle|\, \begin{array}{l} (x,y) \in P(DBNI), \quad (x',y') \in P(CTNI), \\ \sum_i d_i x_{ij} = \sum_i d_i x'_{ij} \quad \forall j, \quad y = y' \end{array} \right\},$$

where the aggregated linear expressions of variables x_{ij}, instead of the individual variables x_{ij}, are duplicated. It then performs Lagrangean relaxation on the copy equalities

$$\sum_i d_i x_{ij} = \sum_i d_i x'_{ij} \quad \forall j \quad \text{and} \quad y = y'.$$

Let $v \in R^m$ and $w \in R^m$ be the corresponding multipliers. We have, after some algebraic manipulation, the following Lagrangean subproblem:

$$\min\left\{ f(x,y) + \sum_j v_j \left(\sum_i d_i x_{ij} \right) + w^T y \mid (x,y) \in P(DBNI) \right\}$$
$$- \max\{(\delta(v) + w)^T y' \mid y' \in P(TI)\},$$

where $\delta(v) \in R^m$ is a column vector whose entries are defined by

$$\delta_j(v) = \max\{0, v_j\} \min\left\{ \sum_i d_i, a_j \right\}, \quad j = 1, \ldots, m.$$

Clearly, the LDA solves Lagrangean subproblems of the same structure as those of the LD. However, the number of multipliers is reduced to $2m$, which may speed up the multiplier adjustment process significantly as the optimization of the Lagrangean dual needs to be performed only in R^{2m}. Let $V(LDA)$ be the dual lower bound generated by the above LDA. Applying Theorem 1, we have the following result:

Corollary 1. $V(LD) = V(LDA)$.

Proof. It has been shown in [5, 11] that

$$\begin{aligned} V(LD) &= \min\{f(x,y) \mid (x,y) \in Co(CTNI) \cap Co(DBNI)\} \\ &= \min\{f(x,y) \mid (x,y) \in D_1\}. \end{aligned}$$

Similarly, one can show ([7]) that

$$\begin{aligned} V(LDA) &= \min\left\{ f(x,y) \,\middle|\, \begin{array}{l} (x,y) \in Co(DBNI), \qquad (x',y') \in Co(CTNI), \\ \sum_i d_i x_{ij} = \sum_i d_i x'_{ij} \quad \forall j, \quad y = y', \end{array} \right\} \\ &= \min\{f(x,y) \mid (x,y) \in D_2\}, \end{aligned}$$

and the result follows directly from Theorem 1. □

In summary, when the LDA approach is applied to CPLP, it generates the same lower bound as the corresponding LD approach. However, substantially fewer multipliers need to be adjusted during the process. Similarly, when the LDA approach is applied to GCPLP and the dynamic production planning model, it also generates the same lower bound as the corresponding LD approach. In addition, one can show that the Lagrangean subproblems solved by the corresponding LD and LDA are the same: for GCPLP, a simple plant location problem and a knapsack problem with an additional cardinality constraint; for the production model, simple plant location problems for each i with “customers” s and “plants” jt, and shortest-path problems for each j.

6. Two identical primal relaxation bounds for nonlinear CPLP

It is well known that the Lagrangean dual of an integer linear program has a primal equivalent in the original space [5]. While one usually finds the Lagrangean lower bound by searching for a best set of Lagrangean multipliers, it is not the only method possible. Michelon and Maculan [15] showed that one can also solve the primal equivalent of the Lagrangean dual by placing the relaxed constraints into the objective function (e.g. using a penalty function method), and then, using a linearization method, such as the Frank and Wolfe algorithm (see [1, p. 580] for a description). As a result, at each iteration, the algorithm solves a linear integer Lagrangean-like subproblem followed by a nonlinear line search. The method is especially useful when the objective function is nonlinear and the nonlinear Lagrangean subproblem is difficult to solve. In this case we ignore the Lagrangean approach and concentrate directly on the primal relaxation (which, by the way, need not be equivalent) and its solution by a linearization method [8]. We now describe two procedures for evaluating primal relaxation bounds for the nonlinear CPLP, and show by applying Theorem 1 that the two bounds are the same.

Consider the following nonlinear CPLP:

$$\min\{f(x,y) \mid (x,y) \in P(DCNIBT)\},$$

where f is a convex nonlinear function of (x, y) and each constraint in $(DCNIBT)$ is as defined in GCPLP. The first primal relaxation approach, called PR1, solve the following penalized problem:

$$\text{minimize} \quad \varphi_1(x, y, x', y') = f(x, y) + \mu_1 \sum_i \sum_j (x_{ij} - x'_{ij})^2 + \rho_1 \sum_j (y_j - y'_j)^2,$$

subject to

$$(x, y) \in Co(CTNI), \quad (x', y') \in Co(DBNI),$$

where μ_1 and ρ_1 are large penalty parameters. Notice that when both parameters approach infinity, PR1 is equivalent to

$$\begin{aligned}
&\min \left\{ f(x, y) \,\middle|\, \begin{array}{l} (x, y) \in Co(DBNI), \quad (x', y') \in Co(CTNI), \\ (x, y) = (x', y'), \end{array} \right\} \\
&\quad = \min\{ f(x, y) \mid (x, y) \in Co(CTNI) \cap Co(DBNI)\} \\
&\quad = \min\{ f(x, y) \mid (x, y) \in D_1\}.
\end{aligned}$$

Let $V(PR1)$ be the optimal objective value of PR1 as both parameters approach infinity. Clearly, $V(PR1)$ is a lower bound for CPLP. The primal relaxation problem can be solved by a linearization method such as the Frank and Wolfe algorithm with a nonlinear line search or simplicial decomposition, which could be more attractive computationally. We describe here the Frank and Wolfe algorithm for simplicity. At iteration k, one has a current iterate (x^k, y^k, x'^k, y'^k) in whose vicinity one creates a linearization $\phi(x, y) + \psi(x', y')$ of the objective function $\varphi_1(x, y, x', y')$. One solves the linearized problem

$$\begin{aligned}
&\min\{\phi(x, y) \mid (x, y) \in Co(DBNI)\} + \min\{\psi(x', y') \mid (x', y') \in Co(CTNI)\} \\
&\quad = \min\{\phi(x, y) \mid (x, y) \in P(DBNI)\} + \min\{\psi(x', y') \mid (x', y') \in P(CTNI)\}.
\end{aligned}$$

The equality is true because both ϕ and ψ are linear functions. The linear subproblem on (x', y') can be further simplified to include only variable y' in the objective and (TI) in the constraint. It is interesting to see that the above linearization subproblem has the same structure as the LD subproblem (for a linear objective function).

Following the same spirit as LDA, we can construct a different primal relaxation, called PR2, for the nonlinear CPLP.

$$\begin{aligned}
\text{minimize} \quad \varphi_2(x, y, x', y') &= f(x, y) + \mu_2 \sum_j \left(\sum_i d_i x_{ij} - \sum_i d_i x'_{ij} \right)^2 \\
&\quad + \rho_2 \sum_j (y_j - y'_j)^2,
\end{aligned}$$

subject to

$$(x, y) \in Co(CTNI), \qquad (x', y') \in Co(DBNI),$$

where μ_2 and ρ_2 are large penalty parameters. When both parameters approach infinity, PR2 is equivalent to

$$\min\left\{ f(x,y) \,\middle|\, \begin{array}{ll} (x,y)\in Co(DBNI), & (x',y')\in Co(CTNI), \\ \sum_i d_i x_{ij} = \sum_i d_i x'_{ij} \quad \forall j, & y = y', \end{array} \right\}$$

$$= \min\{ f(x,y) \mid (x,y)\in D_2 \},$$

Let $V(PR2)$ be the optimal objective value of the above problem as both parameters approach infinity. From Theorem 1, it is clear that the two primal relaxation bounds are the same. That is

Corollary 2. $V(PR1) = V(PR2)$.

In addition, the linearization subproblem of PR2 has the same structure as that of PR1.

References

[1] S.P. Bradley, A.C. Hax, T.L. Magnanti, Applied Mathematical Programming, Addison-Wesley, Reading, MA, 1976.

[2] E. Chajakis, R. de Matta, M. Guignard, A model of dynamic production scheduling with back ordering and Lagrangean approximation schemes, Department of Decision Sciences Report 92-4-1, The Wharton School, University of Pennsylvania, 1992.

[3] G. Cornuejols, R. Sridharan, J.M. Thizy, A comparison of heuristics and relaxations for the capacitated plant location problem, Eur. J. Oper. Res. 50 (1991) 280–297.

[4] R. de Matta, M. Guignard, Dynamic production scheduling for a process industry, Oper. Res. 42 (3) (1994) 1–12.

[5] A.M. Geoffrion, Lagrangean relaxation for integer programming, Math. Programming Study 2 (1974) 82–114.

[6] F. Glover, D. Klingman, Layering strategies for creating exploitable structure in linear and integer programs, Math. Programmming 40 (1988) 165–182.

[7] M. Guignard, General substitution schemes in Lagrangean relaxation: theory and applications, Department of Decision Sciences Report 89-12-07, The Wharton School, University of Pennsylvania, December 1989, latest revision 1/1995.

[8] M. Guignard, Primal relaxations in integer programming, Proc. 7th CLAIO Meeting, Santiago, Chile, 1994.

[9] M. Guignard, E. Chajakis, C. Ryu, Applications of Lagrangean substitution to a hard integer programming problem in forest management, Proc. Int. Symp. on Systems Analysis and Management Decisions in Forestry, Valdivia, Chile, 1993.

[10] M. Guignard, S. Kim, Lagrangean decomposition for integer programming: theory and applications, RAIRO-Oper. Res. 21 (4) (1987) 307–323.

[11] M. Guignard, S. Kim, Lagrangean decomposition: a model yielding stronger Lagrangean bounds, Math. Programming 39 (1987) 215–228.

[12] K. Jornsten, M. Nasberg, P. Smeds, Variable-splitting – a new Lagrangean relaxation approach to some mathematical programming models, Department of Mathematics, Report MAT-R-85-04, Linkoping Institute of Technology, Sweden, 1985.

[13] P. Michelon, Methodes Lagrangiennes pour Programmation Lineaire avec Variables Entieres, Invest. Oper. 2 (1991) 127–146.
[14] P. Michelon, N. Maculan, Lagrangean decomposition for integer nonlinear programming with linear constraints, Math. Programming 52 (1991) 303–313.
[15] P. Michelon, N. Maculan, Solving the Lagrangean dual problem in integer programing (preliminary draft), Report 822, Department of Computer Science and Operations Research, University of Montreal, 1992.
[16] H. Reinoso, N. Maculan, Lagrangean decomposition in integer linear programming: a new scheme, INFOR 30 (1) (1992) 1–5.